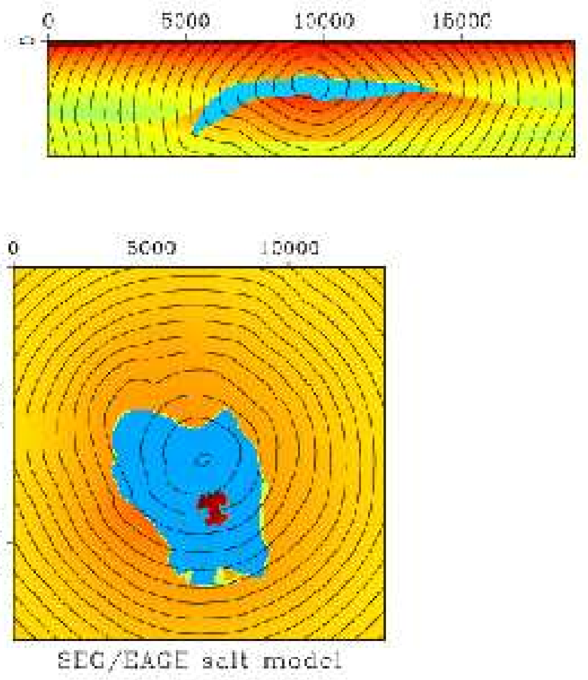
according to the first-order difference operator (1). As a result, the computational error of this method goes to zero with the decrease in the grid size in a linear fashion. The proof of validity of the method (omitted here) is also analogous to that of Dijkstra's algorithm (Sethian, 1996a; Sethian, 1996b). As in most of the shortest-path implementations, the computational cost of extracting the minimum point at each step of the algorithm is greatly reduced [from $O(N)$ to $O(\log N)$ operations] by maintaining a priority-queue structure (heap) for the *NarrowBand* points (Cormen et al., 1990).

Figure 1 shows an example application of the fast marching eikonal solver on the three-dimensional SEG/EAGE salt model. The computation is stable despite the large velocity contrasts in the model. The current implementation takes about 10 seconds for computing a 100x100x100 grid on one node of SGI Origin 200. Alkhalifah and Fomel (1997) discuss the differences between Cartesian and polar coordinate implementations.



salt

Figure 1. Constant-traveltime contours of the first-arrival traveltime, computed in the SEG/EAGE salt model. A point source is positioned inside the salt body. The top plot is a diagonal slice; the bottom plot, a depth slice.

The difference equation (1) is a finite-difference approximation to the continuous eikonal equation

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2 = s^2(x,y,z)\ , \tag{2}$$

where x, y, and z represent the spatial Cartesian coordinates. In the next two sections, I show how the updating procedure can be derived without referring to the eikonal equation, but with the direct use of Fermat's principle.

2005-07-30